What Is Making Artificial Intelligence So Successful Today: A New Generation of Chips, Algorithms, and Toolboxes

Angelos Vlachos, Anastasia Poulopoulou, Christina Giannoula, Georgios Goumas, and Nectarios Koziris

Abstract—**Recent progress in artificial intelligence (AI) is powered by three key elements: algorithmic innovations, specialized chips and hardware, and a rich ecosystem of software and data toolboxes. This paper provides an analysis of these three key elements, tracing the evolution of AI from symbolic systems and small, labeled benchmarks to today's large-scale, generative, and agentic models trained on web-scale corpora. We review the hardware trajectory from central processing units (CPUs) to graphics processing units (GPUs), tensor processing units (TPUs), and custom accelerators, and show how the co-design of chips and models has unlocked improvements in throughput and cost by orders of magnitude. On the algorithmic side, we cover the deep learning revolution, scaling laws, pretraining and fine-tuning paradigms, and multimodal and agentic architectures. We map the modern software stacks, i.e., open-source AI frameworks, end-to-end toolchains, and community datasets, that make model development reproducible and widely accessible. Given the environmental and infrastructural impact of scale, we emphasize the trade-offs in energy, datacenter, and governance. Finally, we identify emerging trends that reshape how AI is developed and deployed.**

Index Terms—**Artificial intelligence (AI), deep learning, chip, algothrithm, toolbox, machine learning, neural network.**

I. Introduction

ARTIFICIAL intelligence (AI) is the capability of computational systems to perform tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making. It is a field of research in computer science that develops methods and software that enable machines to perceive their environment, learn from experience/data, and take actions that maximize their chances of achieving defined goals. AI systems are designed not only to execute instructions but also to adapt to new situations, providing solutions in situations where traditional programming would be insufficient. The widespread applications of AI include advanced web search engines [1], recommendation systems [2], virtual assistants [3], autonomous vehicles [4], generative and creative tools including language models [5] and AI art [6], and superhuman performance in games [7]. Additionally, AI powers specialized domains such as computer vision [8], which enables machines to interpret and analyze images and videos, and natural language processing (NLP) [9], which allows systems to understand, generate, and interact using human language. Many of these applications have become so common and useful that they are no longer perceived as AI by everyday users.

Machine learning (ML), as a central branch of AI, focuses on enabling systems to improve their performance on tasks automatically by learning from data. It has been part of AI from the start and includes several methods. In supervised learning [10], systems are trained using labeled data, where inputs are paired with expected outputs, either for classification (assigning inputs to specific categories) or regression (predicting numeric outputs from numeric inputs). Unsupervised learning [10] analyzes data without labels, identifying patterns and structures to make predictions or detect anomalies. Reinforcement learning [11] teaches systems to make decisions through feedback, rewarding good actions and penalizing poor ones so that the agent gradually learns optimal behaviors. These methods form the backbone of modern AI applications.

Neural networks [12] are a key type of ML model inspired by the structure and function of the human brain. They consist of interconnected layers of "neurons" that process and transform data, allowing the system to model complex, non-linear relationships. Each layer extracts higher-level features from the input, enabling the network to capture intricate patterns. Neural networks are widely used in applications such as image recognition [8], speech processing [13], predictive analytics [14], and language modeling [15], making them a very versatile tool for a variety of AI tasks.

Deep learning (DL) [12] is a specialized subset of neural networks that involves multiple layers, hence "deep" networks, which allow the system to learn hierarchical representations of data. This enables DL models to process high-di-

Manuscript received: November 18, 2025; revised: December 22, 2025; accepted: December 31, 2025. Date of CrossCheck: December 31, 2025. Date of online publication: January 30, 2026.

A. Vlachos, A. Poulopoulou, G. Goumas, and N. Koziris are with the Computer Systems Laboratory, National Technical University of Athens (NTUA), Athens, Greece, and A. Vlachos is also with the Artificial Intelligence and Systems Laboratory NTUA, Athens, Greece (e-mail: aavlachos@cslab.ece.ntua.gr; anastasiapoulop@gmail. com; goumas@cslab. ece. ntua. gr; nkoziris@cslab. ece. ntua.gr).

C. Giannoula (corresponding author) is with the Max Planck Institute for Software Systems, Saarbrücken, Germany (e-mail: cgiannoula@mpi-sws.org).

DOI: 10.35833/MPCE.2025.001111

mensional, unstructured data such as image, audio, video, and text, and to automatically extract relevant features without manual intervention. This enables DL to be effective for many applications that simpler ML methods cannot tackle. DL has powered major breakthroughs in AI, including self-driving vehicles that can interpret complex visual environments [16], NLP systems capable of translation, summarization, or conversation [17], AI tools that generate realistic text or art [5], and industrial systems that detect subtle anomalies [18]. Despite that, large amounts of task-specific data are required in order to be trained properly to achieve these capabilities [19].

In summary, AI is the broadest concept of creating intelligent systems, ML is a data-driven method within AI, neural networks are a key type of ML model, and DL is a powerful extension of neural networks designed to handle complex, high-dimensional data. Understanding this hierarchy helps clarify the capabilities and applications of these technologies and sets a baseline for understanding more intricate AI concepts and ideas mentioned in the following sections.

AI has rapidly evolved from a specialized research field into a foundational technology across many industries. It leads the state-of-the-art applications in many domains such as healthcare, finance, transportation, NLP, computer vision and creativity, and energy. Specifically, for the energy community, the relevance of AI is twofold: it is both a major consumer of computational power [20], and a potential enabler of more sustainable solutions [21]. The following sections address these points.

In this paper, the main goal is to highlight the latest technologies that have driven the rapid rise of AI in recent years, along with applications that AI and ML have enabled. We first present a brief history of DL and how it has evolved, leading up to today's prominent generative AI models such as large language models (LLMs) and diffusion models. We will also delve deeper into the hardware developments that have been crucial in meeting the growing demands of AI research and deployment, and finally look at the current landscape of AI application and discuss what it means for technology and energy consumption.

The objective of this paper is to prepare scientists and engineers who have observed the rapid rise of AI only from a distance, for the challenges and opportunities of the emerging near-artificial general intelligence (AGI) era. By reviewing the history and trends of AI, we aim to show how quickly the research in this field has accelerated, and to highlight how unprepared the society and academic community are for a transformation of this scale. At the same time, by emphasizing the substantial energy consumption associated with training and deploying large-scale AI models, we seek to alert the power and clean energy community to the urgent need for preparation. As AI becomes a driving force across society, we hope to encourage further research into sustainable solutions–efforts that will themselves be supported and accelerated by the very generative AI tools now reshaping the field.

II. Evolution of AI

Figure 1 summarizes the evolution of AI, which serves as the guide of this section. The timeline tracks the progression from early symbolic, rule-based systems to statistical ML and DL. These advancements have enabled the development of generative AI and LLMs. The timeline ends with the current transition to agentic AI, which extends the capabilities of LLMs from content generation to autonomous reasoning and task execution.

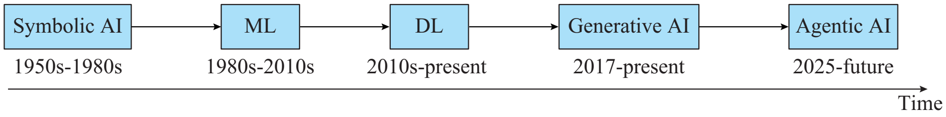

Fig. 1. Evolution of AI.

A. From Symbolic Expert Systems to Statistical ML

The earliest developments in AI were dominated by symbolic methods, most notably expert systems, which relied on explicitly encoded rules and logical reasoning [22]. While these systems achieved success in highly constrained domains such as medical diagnosis [23] and equipment troubleshooting, their scalability and adaptability were limited. The 1990s and 2000s witnessed a paradigm shift toward data-driven methods, as statistical ML emerged as a more flexible and generalizable alternative [24]. Unlike symbolic systems, statistical methods leverage large datasets and probabilistic models, enabling broader applications such as pattern recognition, speech processing, and predictive modeling. This transition sets the foundation for the current wave of AI, which couples powerful algorithms with advances in computing hardware and specialized frameworks, paving the way for the large-scale and diverse applications observed nowadays.

B. Revolution of DL: Neural Networks, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Graph Neural Networks (GNNs)

DL [12] refers to neural network models with multiple layers that can automatically learn the hierarchical representations of data. Its rise is attributable to improvements in training algorithms (e.g., back-propagation [25]), activation functions [26], availability of large datasets, and hardware capable of massive parallel computation. The evolution of deep architectures has followed a few key threads.

1) Deep feed-forward neural networks. Early multi-layer perceptrons laid the groundwork, but suffered from limitations such as vanishing gradients and high computational cost, as they connected all inputs with all outputs.

2) CNNs. Originally inspired by neuroscientific models, e.g., Neocognitron by Fukushima [27], CNNs became practical in vision tasks (i.e., image classification and object detec-

tion), especially after the success of models in [28] and AlexNet [29], etc. Their spatial hierarchy, parameter sharing, and pooling operations make them extremely effective and efficient for grid-structured data.

3) RNNs and variants. Designed for sequential or temporal data, RNNs (especially long short-term memory (LSTM) [14] and gated recurrent units (GRUs) [30]) have enabled the modeling of speech, language, and time-series data, including domains where past inputs influence future ones.

Another dimension of revolution is the scaling of data, models, and compute. After early success, deep networks have grown in depth, width, and complexity, e. g., ResNets [31], Inception [32], and become feasible by algorithmic tricks (residual connections and batch normalization [33]) as well as hardware advances.

With the success of DL came new applications and industrial adoption across multiple domains. In computer vision, DL now sets the state-of-the-art in object detection, segmentation, and classification, with CNNs powering many real-world systems, including face recognition [8] and medical imaging [34]. For sequence and time-series data, RNNs, LSTM, and GRU networks are widely used in applications such as speech recognition [35], machine translation [36], demand forecasting, and anomaly detection. Meanwhile, GNNs have found application in several key areas: recommender systems [37], modeling of complex networks (e. g., social and communication), chemistry through molecular graphs [38], and, more recently, infrastructure systems.

C. Generative AI

In recent years, the frontier of AI has been dominated by generative AI, a subfield of AI that uses generative models to produce new content such as text, image, music, or code by learning from existing data. Unlike models mentioned so far that essentially analyze data, generative AI models mimic human creativity, producing outputs that are sometimes indistinguishable from human-created work, in response to a user's prompt. The development of modern LLMs is largely founded upon the transformer architecture introduced in 2017 [15], followed by the introduction of bidirectional encoder representations from transformers (BERT) in 2019 [9], which pioneers bidirectional pre-training for language understanding, and subsequent milestones including the development of the generative pre-trained transformer (GPT) series such as GPT-3 in 2020 [5].

A key stepping stone in combining vision and language is contrastive language-image pre-training (CLIP) in 2021 [39], which learns joint representations of images and text via large image-caption datasets and a contrastive learning objective, enabling the use of visual modalities as inputs too, additionally to text. For image and other media generation, a key breakthrough has come with diffusion models, which are fundamentally formalized by studies such as denoising diffusion probabilistic models in 2020 [40], and later made practical for high-resolution synthesis through latent diffusion models in 2022 [41]. The latest trend involves multimodal AI (e.g., Gemini [42], GPT-4V [43]), which integrates and processes information from different data types such as text and images. These models can be developed at massive parameter sizes, trained on enormous datasets, and operated using vast compute infrastructures. In this subsection, we further discuss some unique characteristics of the generative era, e. g., how they are trained, scaled, and become multimodal.

1) Pre-training, Post-training, and Inference

In addition to architecture and scale, generative and agentic models undergo distinct phases in their lifecycle: pre-training, post-training, and inference. Pre-training is the initial phase in which a model is exposed to massive amounts of data in order to learn general representations of language, vision, or other modalities. Post-training (also called fine-tuning) refines the pre-trained model using labeled or more narrowly focused and curated data, so that it performs well on specific tasks, styles, or domains. One very common scenario of post-training in the case of LLMs is instruction tuning [44], [45], where the model is fine-tuned in specific instruction-following data to enhance its chatbot-like capabilities. In the more recent era of LLMs, a further phase of alignment fine-tuning has emerged, namely reinforcement learning from human feedback (RLHF) [46]. RLHF uses human preference judgments to guide the model's policy via reinforcement learning so that its outputs better align with human value, intent, or a specific task. The integration of RLHF has become central in many agentic and interactive LLM systems, often providing stronger behavioural alignment than supervised fine-tuning alone.

Finally, inference is the stage when the model is deployed in real use. Given new inputs (prompts), the model generates outputs using its learned parameters without further changing them. Each phase has different computational, data, energy demands, and trade-offs. Pre-training tends to be computationally expensive, but is performed only once, while post-training is a less expensive and shorter process that can be performed repeatedly for task-specific adaptation. Although a single LLM inference operation is fast and inexpensive, the repeated nature of its execution for every user's prompt means that its operational expense scales with the usage volume, leading to cumulative inference cost to ultimately exceed the initial one-time investment required for pre-training.

2) Scaling of Parameters, Training Corpus, and Compute

The defining characteristic of this era is exponential scaling across three dimensions: the number of parameters, the size of the training corpus, and the compute required for training and inference [47], [48].

1) Parameters: early foundational models had modest parameter counts. GPT-1 in 2018 featured approximately 117 million parameters [17]. BERT Large had around 340 million parameters [9]. The leap to GPT-3 in 2020 marked a monumental jump to roughly 175 billion parameters [5]. Contemporary models, while exact figures are often proprietary, are rumored to lie in the trillion-parameter regime, including variants of GPT-4, Llama 4 Behemoth, DeepSeek V3, and Qwen3-Max.

2) Training corpus: GPT-3 was trained on roughly 300 billion tokens, but state-of-the-art models now leverage the orders of magnitude of data to be larger. For instance, DeepSeek V3 [49] reportedly used 14.8 trillion tokens, while Lla-

ma 4 [50] and Qwen3-Max [51] are said to incorporate 30-36 trillion tokens in training.

3) Compute: the energy and hardware demands need to scale accordingly. The pre-training of GPT-$3 3.14\times 10^{23}$ floating-point operations per second (FLOPs) ($\approx$3640 PetaFLOP-days (PF-days)). Modern efforts deploy a massive number of graphics processing units (GPUs): DeepSeek V3 used 2048 NVIDIA H800 GPUs over 2.788 million GPU-hours. The more ambitious Llama 4 Behemoth project is rumored to have used up to 32000 GPUs for pre-training. These figures underscore not only the computational intensity but also the immense energy, cooling, networking, and logistical overheads entailed by training and deploying models at this scale.

3) In-context Learning

An important ability in current generative AI, particularly within LLMs, is the emergence of "in-context learning". Unlike traditional supervised learning, which requires updating model weights based on vast labeled datasets, in-context learning allows models to perform tasks based on prompts provided at the inference time. This leads to two distinct capabilities: zero-shot learning (ZSL) and few-shot learning (FSL). ZSL refers to the ability of model to tackle a novel task without seeing any specific examples, relying solely on a natural language description of the task [52]. FSL extends this by providing the model with a small number of demonstrations (e.g., one to five examples) within the input context to guide its output [53]. In the context of power systems and energy grids, these paradigms are not merely convenient features; they are crucial for addressing two fundamental challenges: data scarcity for critical events and computational sustainability. Specifically, these methods provide a robust solution to the inherent lack of labeled data for rare, high-impact anomalies such as cascading blackouts or cyber-physical attacks [54], where it is difficult to construct balanced training sets [55]. By leveraging pre-trained knowledge and few examples to identify these events without retraining, they bypass the need for massive datasets. Furthermore, utilizing models in a "frozen" state rather than performing computationally expensive full fine-tuning could significantly reduce the energy overhead, but slightly increase the cost of inference, i.e., in the case of a few-shot scenario.

4) Multimodal Generative Models

While text-based LLMs remain a central pillar, generative AI has evolved to embrace multimodal architectures. Using architectures such as CLIP [39] has enabled the use of visual modalities (e.g., images or videos) as inputs along with text, which enables a plethora of new applications such as in visual reasoning and understanding, healthcare and medical diagnostics, or robotics and autonomous systems. Diffusion models (e.g., stable diffusion [41], [56], DALL·E [57]) now enable high-fidelity image generation, image-to-image translation, and combinations of modalities (vision + language). Multimodal generation models must contend with not only scaling in parameters and data, but also in the complexity of inference. Generating a high-resolution image/video or analyzing multiple images typically incurs more energy and latency compared with processing or generating textual output.

D. Agentic AI: Beyond Generation to Autonomy

Generative models have enabled powerful capabilities in content synthesis, completion, and cross-modal generation. Building on this foundation, traditional AI agents act as modular, task-specific systems that augment language or image models with tools, prompts, or scripted behaviors. The next frontier, agentic AI, goes further by planning and acting autonomously over multiple steps to achieve complex goals. These systems combine multi-agent coordination, dynamic task decomposition, persistent memory, and orchestrated autonomy, allowing them to adapt and make decisions with the minimal human intervention in evolving environments. [58], [59]. In practice, agentic AI interprets natural language goals, decomposes them into subgoals, plans sequences of actions, invokes external tools or application programming interfaces (APIs), observes outcomes, and could replan when necessary. It thus bridges the gap between high-level intent and real-world effect [60]. Architecturally, it often adopts orchestrator-agent or multi-agent frameworks, where separate agents specialize (e.g., search, reasoning, tool execution) under coordination.

Agentic AI is important for several reasons. First, it enables more advanced automation of workflow. Instead of handling isolated tasks, it can carry out complete processes that include data collection, reasoning, tool use, and result generation with the minimal oversight by humans. Second, it reduces the human cognitive load by embedding planning, coordination, and handling of errors directly into the agent, rather than relying on humans to manage each subtask. Third, it enables scalable autonomy, meaning that multiple agents can collaborate to address complex problems that exceed the capabilities of a single agent. Current applications of agentic AI span fields including healthcare, finance, robotics, adaptive software, and autonomous scientific research [61].

Yet, agentic AI comes with challenges. Reliability and robustness remain the major concerns. The agents may drift, enter loops, or propose unsafe actions. Coordination failures or error propagation across agents can lead to cascading failures [62], [63]. Scalability is another issue. As agent counts increase, communication overhead and orchestration complexity grow nonlinearly. The scope of task is also constrained. In many empirical settings, agentic AI performs best when tasks remain within moderate complexity or time horizons. Furthermore, the computational and energy demands of planning, environment simulation, tool invocation, and error correction loops are substantial and compound the costs of generative backbones. Finally, governance, oversight, alignment, and transparency must be built into the architecture to ensure safe deployment of agentic AI.

III. Hardware: Engine That Powers AI

A. From Central Processing Units (CPUs) to GPUs

CPUs represent the primary computational system engineered to efficiently execute sequential instruction streams across a wide range of application domains including databases, data analytics, scientific simulations, financial modeling, and interactive user applications. They have been optimized to minimize data access latency and maximize single-threaded performance through sophisticated microarchitectural techniques including branch predictors, prefetchers, out-of-order execution (dynamically reordering instructions to reduce pipeline stalls), and multi-level cache hierarchies that keep frequently accessed data close to the processing cores [64]. While modern CPUs incorporate parallelism through multiple cores, typically ranging from 4 to 64 cores in consumer systems and up to hundreds of cores in high-end server processors, this parallel capability remains fundamentally limited compared with specialized architectures. CPUs excel in applications requiring complex control flow, irregular memory access patterns, and low-latency responsiveness. Future microarchitectural innovations continue to prioritize latency reduction and single-thread performance optimization over massive parallelism, maintaining the role of CPUs as the backbone of general-purpose computing systems [64].

GPUs have revolutionized parallel computing by providing significantly more parallelism than CPUs. They were originally designed for graphic applications and gaming workloads that require rendering thousands of pixels simultaneously. They feature hardware-managed caches and massive parallelism with thousands of cores organized in a single-instruction multiple-thread (SIMT) execution model, enabling efficient processing of regular parallel computations [64]. Early GPUs demonstrated impressive performance potential in gaming applications, prompting GPU manufacturers to quickly modify their microarchitectures for general-purpose GPUs (GP-GPUs). GPUs introduce double-precision floating-point support, enhancing the instruction sets for mathematical operations, and improving memory controllers to support diverse applications ranging from scientific computing to cryptographic processing [65]. Moreover, GPUs have been specifically designed to provide substantially higher memory bandwidth than CPUs, typically delivering five to ten times more bandwidth. This is because their thousands of parallel cores necessitate massive data throughput to maintain the computational efficiency [66]. However, early GPU generations consumed significantly more energy, often requiring two to three times than CPUs. Recent architectural advances have focused significant efforts on improving their performance-per-watt ratios, making them increasingly attractive for a wide range of applications [67].

GP-GPUs have also enabled significant advancements in programmability. Early GPUs were limited to fixed-function graphic pipelines, creating programming burdens for non-graphic computations. However, high-level programming languages such as NVIDIA's compute unified device architecture (CUDA) [68] and the open standard OpenCL [69] have fundamentally transformed GPU accessibility, providing developers with familiar C-like programming models and comprehensive software development kits [70], [71]. These programming frameworks abstract the underlying hardware complexity while exposing fine-grained control over thread execution, memory hierarchy management, and inter-core communication. Additionally, GPU vendors have invested heavily in optimizing compilers, debugging tools, and performance profilers, while introducing automatic optimization techniques that significantly reduce the programming complexity barrier, thereby making the computation of GP-GPUs accessible to a broader range of developers [71].

ML and DL workloads, henceforth named as AI workloads, inherently fit well on GPUs due to their embarrassingly parallel computational characteristics. They primarily consist of massive matrix-matrix multiplications, element-wise operations, and linear algebra kernels that exhibit regular data access patterns [72], [73]. GPUs provide the essential massive parallelism needed for training and serving ML models. To support the thousands of simultaneous computations executed by GPU cores, GPUs have been enhanced with advanced memory technologies including three-dimensional (3D) memory stacking techniques and high-bandwidth memory (HBM) [74]. This architectural alignment has resulted in significant performance benefits for AI workloads, and given the growing commercial importance of AI applications, GPU architects have transformed GPU microarchitectures to more effectively accommodate AI-specific computational requirements. Specifically, GPUs have been enhanced with specialized tensor cores [75] designed to accelerate matrix operations, alongside support for mixed-precision formats that balance the computational throughput with numerical accuracy. GPU manufacturers have progressively expanded numerical format support beyond traditional 32-bit floating-point (FP32) to include 16-bit floating-point (FP16) for memory efficiency, 16-bit brain floating-point (BF16) for improved numerical stability, and lower precision formats such as 8-bit integer (INT8), 4-bit integer (INT4), and recently 4-bit floating-point (FP4). Additionally, modern GPUs incorporate advanced interconnect technologies such as NVLink and optimized multi-GPU communication protocols to enable efficient multi-GPU parallelism [65]. As increasingly large DL models such as LLMs cannot fit within single GPU memory constraints due to their substantial parameter counts, efficient multi-GPU communication becomes essential for high-performance distributed training, fine-tuning, inference, and serving across multiple GPU devices [76]. Complementing these hardware advances, the software ecosystem has matured significantly with comprehensive development frameworks and sophisticated compilers that automatically optimize AI workloads for GPU architectures, thereby maximizing hardware utilization for AI applications [77], [78].

Although GPUs enable high performance in AI workloads, the latest high-end GPU architectures are extremely power-hungry, consuming substantial energy and contributing significantly to carbon emissions due to their design characteristics including thousands of parallel cores operating at high frequencies and extensive memory bandwidth capabilities [67]. Generative AI models, which contain billions to trillions of parameters, have necessitated the deployment of these energy-intensive GPUs, and, more importantly, require multi-GPU configurations for training and inference, exponentially increasing energy consumption and carbon emissions [79]. Architectural advancements such as specialized compute units such as Tensor Cores inherently result in elevated energy densities and thermal design challenges that further exacerbate energy consumption. Consequently, the energy efficiency and carbon footprint reduction have emerged as critical microarchitectural design challenges that must be addressed to ensure the sustainable development of AI [80].

B. Tensor Processing Units (TPUs) and AI Accelerators

AI accelerators represent a new generation of specialized processors designed exclusively for AI computations, emerging as purpose-built alternatives to GPUs. While GPUs have demonstrated significant improvements of performance over CPUs for AI workloads due to their parallel architecture, they were originally designed for graphics rendering and carried substantial overhead from graphic-specific hardware characteristics that were unnecessary for AI computations. To this end, technology companies have developed dedicated AI accelerators that aim to optimize multiple aspects of the chip architecture specifically tailored for AI, including specialized matrix multiplication units, optimized memory hierarchies for access patterns to AI data, custom numerical precision formats, and streamlined instruction sets tailored for tensor operations. These AI accelerators typically feature large on-chip memory to minimize the expensive off-chip data movement, systolic array architectures for efficient matrix computations, and custom interconnects designed for multichip scaling of large AI models. Unlike GPUs that must balance graphics and compute workloads, AI accelerators achieve significantly higher performance per watt and faster training and inference time by dedicating their entire silicon area and design complexity exclusively to AI algorithms.

TPUs represent the first commercially available AI accelerators, introduced in 2016 as purpose-built processors for neural network computations [81]. TPUs employ a systolic array architecture, where data flow through a grid of processing elements in a synchronized manner. TPUs enable highly efficient matrix-matrix multiplication operations that are fundamental computational kernels in AI workloads. Originally, TPUs are designed specifically for CNNs and inference workloads, featuring 8-bit integer arithmetic and large on-chip memory to minimize data movement costs. Subsequent generations (TPU V2, V3, and V4) have expanded their capabilities to support training workloads, floating-point precision, and diverse AI model architectures including transformers and RNNs [82]. Compared with GPUs, TPUs achieve superior performance per watt and faster execution time for AI workloads by optimizing data flow based hardware support for tensor operations.

Apart from Google, other major cloud providers have also developed custom AI accelerators to optimize performance and reduce costs for their specific infrastructure and AI workloads. Specifically, Amazon Web Services (AWS) offers two types of AI accelerators: ① trainium chips for training, and ② inferentia chips for inference, featuring custom instruction sets, high-bandwidth memory, vector processing engines, and optimized interconnects designed to integrate seamlessly with AWS cloud services [83], [84]. Similarly, Microsoft's Maia accelerators employ co-optimization of both hardware and software, which is specifically tailored for Azure's AI services, featuring advanced memory hierarchies and interconnects designed for large-scale distributed training [85]. These cloud-native AI accelerators offer significant advantages including lower total cost of ownership, optimized performance for cloud-specific AI workloads, seamless integration with cloud services and frameworks, and the ability to scale efficiently across thousands of accelerators in datacenter deployments.

Specialized AI accelerator companies have developed innovative architectures targeting different aspects of AI performance. Graphcore's intelligence processing units (IPUs) include thousands of high-performance parallel cores, where each core and its locally accessible in-processor memory unit form a tile, and data are exchanged among tiles using a bulk synchronous parallel model, enabling efficient model parallelism for large AI models [86]. Cerebras' wafer-scale engine (WSE) represents an extreme method by utilizing an entire silicon wafer as a single chip with over 800000 cores and 40 GB of on-chip memory, delivering hundreds of PetaFLOPs of AI compute throughput and enabling unprecedented parallelism for training massive AI models [87]. Intel's Gaudi accelerators focus on scalable training through Ethernet-based interconnects and mixed-precision capabilities, offering flexible deployment options, cost-effective scaling, and generality in supporting a wide variety of AI models and frameworks [88]. These specialized AI architectures provide key benefits including significantly reduced memory bottlenecks, higher computational density, improved energy efficiency, and optimized performance for specific AI model types and scaling scenarios.

Lastly, neural processing units (NPUs) represent a distinct category of AI accelerators characterized by their integration into system-on-chip (SoC) architectures and optimization for diverse deployment scenarios. NPUs feature specialized AI instruction sets, variable precision arithmetic support, and energy-efficient designs that consume significantly less energy than CPUs or GPUs, while being highly optimized for matrix-matrix and matrix-vector computational kernels [89]. Examples include Apple's Neural Engine [90] integrated into mobile and desktop processors, Qualcomm's Hexagon NPUs in Snapdragon SoCs [91], and dedicated edge devices such as Google's Edge TPU [92]. NPUs work alongside CPUs and GPUs to handle AI workloads, which are ideal for mobile and battery-powered devices. Unlike larger AI accelerators, NPUs prioritize energy efficiency and real-time processing capabilities, making them particularly suitable for edge computing applications including smart phones, Internet of Things (IoT) devices, autonomous vehicles, and embedded systems where energy constraints and latency requirements are critical, while maintaining sufficient computational capability for AI inference and lightweight training tasks.

C. Specialized Hardware: Application-specific Integrated Circuits (ASICs), Field-programmable Gate Arrays (FPGAs), and Neuromorphic Computing

Apart from GPUs and AI accelerators discussed above, ASICs are more specialized for AI acceleration. ASICs are built for specific workloads, and feature custom silicon designs optimized for specific neural network architectures and AI applications. Unlike general-purpose AI accelerators, ASICs are designed from the ground up for the particular AI workloads. Since they have dedicated hardware designs for target AI workloads, they consistently outperform more general-purpose accelerators (e.g., CPUs or GPUs) in both performance and energy comsuption, as they enable extreme optimization by eliminating unnecessary functionality and hardwiring specific computational patterns directly into silicon [93]. ASICs serve two primary domains: large-scale training and inference in datacenters, and resource-constrained edge AI applications [94]. Notable examples include Google's TPUs for cloud-scale operations, along with other specialized architectures such as NPUs, IPUs, and WSEs. In the edge computing space, Apple's Neural Engine and Huawei's Ascend processors demonstrate how ASICs enable on-device AI capabilities while maintaining strict power and thermal constraints. A few other examples include Tesla's full self-driving (FSD) [95] chip optimized specifically for computer vision in autonomous vehicles, Meta's Training and Inference Accelerator (MTIA) [96] designed for their specific recommendation and language models, and Hailo's AI processors targeting edge inference applications. ASICs achieve superior performance per watt and cost efficiency for their target applications by dedicating every transistor to specific computational requirements, featuring custom datapaths, optimized memory hierarchies, and specialized arithmetic units that perfectly match the target neural network operations. Thus, ASICs are well-suited in application-specific and high-volume deployment scenarios, where workload characteristics are well-defined and stable.

Although ASICs enable high performance, they can be difficult to program and modify once fabricated. FPGAs offer an attractive trade-off between the flexibility of software-based development and the high performance of custom hardware. Unlike general-purpose GPUs with fixed architectures that cannot be reprogrammed, FPGAs feature reconfigurable logic blocks that can be programmed to implement custom digital circuits optimized for specific AI workloads. FPGAs provide reconfigurability that enables application-specific optimization, resulting in reduced latency and energy consumption. Designers can implement domain-specific optimizations and tailor circuits to specific workloads, achieving exceptional performance per watt [97]. FPGAs are typically used in AI applications that require ultra-low latency such as autonomous vehicle control systems and edge inference, where fast response time in miliseconds is critical, and they also excel in edge computing and IoT applications. Major FPGA vendors such as Intel (formerly Altera) and AMD Xilinx have developed AI-specific architectures, e.g., the Versal Adaptive Compute Acceleration Platform (ACAP) [98], which combines traditional FPGA fabric with dedicated AI engines, high-bandwidth memory interfaces, and integrated processing cores. Microsoft's Project Brainwave [99] exemplifies by deploying FPGAs across Azure datacenters to accelerate AI inference services with consistently low latency, but FPGAs are also used by other cloud providers (e.g., Micr511 AWS, Huawei, and Baidu) for scalable AI inference. The key advantage of FPGAs lies in their ability to implement custom precision arithmetic formats, create optimized dataflow architectures that minimize memory access overhead, and adapt to evolving AI algorithms through reconfiguration [97]. This makes them particularly well-suited for AI applications where computational requirements may change over time, such as low-latency inference in real-time AI applications.

Neuromorphic computing is fundamentally different for AI processing, as it mimics the structure and operation of biological neural networks (i.e., the way human brains operate) through event-driven, asynchronous computation. This contrasts with the synchronous clock-based operation of traditional digital processors. These neurological and biological mechanisms are modeled through spiking neural networks (SNNs), which are composed of spiking neurons and synapses that replicate human brain's event-driven signaling, resulting in sparse and asynchronous computation [100], [101]. Instead of following the traditional von Neumann architectures, where computation and memory are physically separated, neuromorphic systems integrate processing and memory in a highly parallel, event-driven manner [102]. As computations occur only when spikes are present, neuromorphic systems achieve extremely low energy consumption relatively to all the accelerators described above. By both storing and processing data within individual neurons, they deliver lower latency and faster computation compared with von Neumann architectures. Neuromorphic chips such as Intel's Loihi 2, IBM's TrueNorth, and BrainChip's Akida implement SNNs where information is encoded in the timing and frequency of discrete events (spikes) rather than continuous numerical values, enabling ultra-low energy consumption for specific AI tasks such as pattern recognition, sensory processing, and adaptive learning [100]. In academia, early implementations include Stanford University's Neurogrid [103], a mixed analog-digital multichip system capable of simulating a million neurons with billions of synaptic connections in real time. Research hub Interuniversity Microelectronics Centre (IMEC) developed a self-learning neuromorphic chip, while the European Union's Human Brain Project [104] produced large-scale neuromorphic machines. The neuromorphic systems can be well-suited for applications that require real-time processing with the minimal energy consumption such as robotics control and brain-computer interfaces, achieving higher energy efficiency by several orders of magnitude compared with traditional processors for certain workloads. However, neuromorphic computing still remains largely experimental.

D. Datacenters and Supercomputing Infrustructure for AI at Scale

State-of-the-art AI workloads have evolved to unprecedented computational and memory requirements by incorporating billions to trillions of parameters to achieve high-quality performance. This trend necessitates immense compute and memory capabilities that far exceed those of single computing nodes, requiring massive datacenter and supercomputing infrastructure to support AI training and deployment. LLMs such as GPT-4 and Llama require hundreds to thousands of high-end GPUs for training and hundreds of GPUs even for inference workloads, consuming PetaFLOPs of computational resources, generating terabytes of intermediate data, and consuming megawatts of power. These models demand enormous computational throughput alongside high-bandwidth memory systems and networks, as well as advanced storage technologies to manage massive datasets and model checkpoints [105]. This emerging trend has fundamentally transformed AI development from single-node computing to complex distributed systems, where thousands of accelerators must be coordinated across datacenter networks with sophisticated interconnects, making supercomputing infrastructure an essential prerequisite for advancing the frontiers of AI research and deployment.

Datacenters have undergone fundamental transformations to accommodate the demanding requirements of AI workloads. They have evolved from traditional server architectures to heterogeneous computing environments that integrate diverse AI accelerators including GPUs, TPUs, and specialized AI chips with massive compute and memory capabilities, featuring heterogeneous hardware characteristics to accommodate different types of emerging AI workloads, with all these nodes interconnected over high-bandwidth networks within and across computing nodes [106]. The power and thermal challenges are unprecedented, with individual GPUs consuming 250-700 W power and AI clusters requiring thousands of these accelerators, necessitating megawatts of power delivery and sophisticated cooling solutions, alongside specialized power infrastructure featuring redundant power supplies, advanced power distribution units, and backup systems to ensure continuous operation. To address the requirements of latency and performance, these heterogeneous computing nodes have been integrated with high-bandwidth networking architectures featuring InfiniBand EDR/HDR/NDR (100/200/400 Gbit/s), high-speed Ethernet (100 GbE/400 GbE), and custom interconnect solutions organized in multi-tier topologies designed to minimize communication bottlenecks during distributed training and serving [106], [107]. Additionally, fast interconnection technologies are integrated within computing nodes, including NVLink 4.0 [108] providing 900 GB/s bidirectional bandwidth between GPU accelerators, peripheral component interconnect express (PCIe) 5.0 interfaces, and emerging compute express link (CXL) [109] technology that enables cache-coherent memory sharing and pooling across heterogeneous processors within a node. To further enhance the compute and communication throughput, manufacturers have developed dedicated switches and network devices such as NVIDIA's Quantum InfiniBand switches and Mellanox Spectrum Ethernet switches [107], some of which integrate lightweight cores and perform the computation as data are being exchanged among AI accelerators. Finally, these AI facilities integrate advanced storage systems including parallel file systems (e.g., Lustre file system, IBM Spectrum Scale, BeeGFS) [110], [111] and high-performance object storage to enable efficient access to massive datasets measured in petabytes for AI training workloads, while providing the necessary input/output (I/O) throughput to supply AI accelerators with data efficiently, without pipeline bottlenecks that would underutilize expensive computational resources.

Supercomputers designed specifically for AI workloads represent the pinnacle of distributed computing systems, featuring massive clusters of specialized accelerators optimized for neural network training and inference at unprecedented scales. Notable AI supercomputers include NVIDIA's DGX SuperPOD systems such as the DGX A100 SuperPOD with up to 140 computing nodes containing 1120 A100 GPUs interconnected via NVLink and InfiniBand HDR, delivering more than 400 PetaFLOPs of AI performance [112]. Google's TPU Pods [113] featuring up to 4096 TPU V4 chips in a single Pod with 1.1 exaflops of computational capacity, and Microsoft's supercomputing infrastructure built on Azure with over 14400 H100 GPUs is specifically designed for training LLMs [106]. These systems feature sophisticated multi-node architectures with hundreds to thousands of computing nodes, hierarchical memory systems spanning the high-bandwidth memory of GPU (up to 80 GB HBM2e per GPU), node-level double data rate (DDR) memory (up to 2 TB per node), and shared parallel storage systems measured in petabytes. Advanced fault tolerance mechanisms include checkpoint and restart capabilities, redundant networking paths, and proactive hardware monitoring to maintain reliability across millions of components. Furthermore, these supercomputers employ specialized job scheduling systems such as Slurm and Kubernetes with AI-aware resource management that optimize the allocation of GPUs, and provide efficient resource sharing among concurrent AI workloads while managing energy consumption and thermal constraints across the entire facility.

Moreover, hyperscale cloud providers have established global networks of specialized datacenters to serve the computational demands of leading AI companies and researchers worldwide. Major cloud providers offer dedicated AI infrastructure including AWS with EC2 P4d instances [114] featuring 8 NVIDIA A100 GPUs per node and P5 instances [115] with H100 GPUs, supporting companies such as Anthropic for Claude model training, Google Cloud Platform for providing TPU pods used by organizations for LLM development, and Azure for offering NDv2/NDv4 instances [116] with up to 8 V100/A100 GPUs per node, while hosting GPT models of OpenAI through their strategic partnership and dedicated supercomputing infrastructures. Specialized AI cloud providers such as CoreWeave, Lambda Labs, Paperspace, and RunPod have emerged to offer GPU-focused infrastructure with competitive pricing and AI-optimized configurations, often providing faster deployment and more flexible resource allocation than traditional cloud giants. These cloud platforms deliver the infrastructure as a service (IaaS) capabilities including on-demand GPU clusters that can scale from single instances to thousands of accelerators within minutes. The geographic distribution of cloud AI infrastructure spans multiple continents with strategically located datacenters in North America (AWS US-West-2, Google US-Central1), Europe (EU-West, EU-Central), and Asia-Pacific (Asia-Southeast, Asia-Northeast) regions, enabling AI companies to deploy models closer to end-users for reduced latency while serving millions of users globally.

E. Energy Consumption and Environmental Footprint of AI

The rapid exponential scaling of contemporary AI models, from billions to trillions of parameters, together with the concomitant growth in dataset size and continuous large-scale service deployment, has induced a distinct and rapidly growing set of energy and infrastructure challenges. Large cloud datacenters that host training and serving workloads are dominated by energy-intensive accelerators and substantial facility support systems (cooling, power distribution, networking, and storage). These facilities therefore generate two conceptually distinct classes of carbon emissions that must be accounted for in any comprehensive environmental assessment: embodied emissions incurred during the manufacture, transport, and end-of-life disposal of hardware components, and operational emissions incurred by the energy consumed in training, inference, and the ancillary facility systems required to keep the hardware and services available [105]. At the scale of modern AI services, these contributions are nontrivial: analyses of high-volume conversational systems estimate the daily energy consumption on the order of that consumed by hundreds of thousands of households (for example, energy demand comparable to the daily electricity usage of approximately 180000 households), and information and communication technology (ICT) is projected by sectoral analyses to account for a significant portion of global emissions over the next decade. These magnitudes emphasize that AI growth is not only a computing or economic problem, but a systemic challenge for power system sustainability [105], [117].

Energy consumption is concentrated in two lifecycle phases, i. e., training and inference, but their relative importance depends strongly on the deployment scale and use patterns. Training very large models remains extraordinarily energy-intensive; however, when models are widely deployed for inference, the aggregate energy consumed by serving can dominate the lifecycle of a model. Recent empirical studies indicate that architectural and training environment choices can produce very large reductions in training energy (e. g., [118] reports reductions on the order of 80.7% under certain optimizations with only negligible loss in task correctness), whereas detailed measurements of inference on modern accelerators reveal that per-query energy is sensitive to model architecture and sequence length. For example, measurements on frontier models running on H100-class hardware yield median per-query energies on the order of 0.34 Wh under typical token lengths, rising to 4.32 Wh for token lengths increased by 15 times. Extrapolating these per-query costs to population-scale workloads produces daily energy demands on the order of 10^8-10^9 Wh [119]. Under realistic conditions, serving 1 billion queries per day has been estimated to require on the order of 0.8 GWh energy. Workloads with a non-negligible fraction of much longer queries (e. g., 10%) could raise that figure toward 1.8 GWh per day absent countervailing efficiency improvements, though modest system and software efficiency gains can materially reduce these totals [119]. These quantitative examples illustrate how modest changes in per-query energy or fraction of long queries could translate into very large shifts in absolute energy demand once services reach the global scale [80]

Beyond the energy consumption by direct computing, facility-level overhead significantly amplifies the lifecycle energy and emissions. Cooling, uninterruptible power supplies, power distribution losses, networking equipment, and storage subsystems impose additional loads that are captured, albeit imperfectly, by facility metrics such as power usage effectiveness (PUE). In inefficient facilities, these overheads can effectively double or triple the energy attributable to raw compute, so careful attention is indispensable for facility design, siting, and operations. Macro-scale data contextualize the sectoral impact: recent assessments place the energy consumption of datacenters in the hundreds of terawatt-hours per year (for example, IEA/Nature [120] estimates for 2022 are in the range of 240-340 TWh, roughly 1%-1.3% of global electric power demand), underscoring that trends in AI deployment will interact materially with broader efforts in power system planning and decarbonization.

As the environmental consequences of AI are inherently coupled to power system operations and long-term infrastructure planning, it is vital to point out the importance of transparent, lifecycle-aware reporting and of coordinated system planning. To be both standardized and reproducible, energy accounting must ideally incorporate measured per-query energy, PUE-normalized facility loads, embodied emissions (amortized using realistic lifecycle assumptions), and the carbon intensity of purchased as well as onsite electricity. It will enable objective comparison of architectural choices and deployment strategies, and will allow researchers and power system operators to evaluate the system-level implications of shifts in AI demand. Mitigation is therefore not solely a matter of hardware design or algorithmic optimization, but requires alignment across hardware innovation, datacenter engineering, model development, operational policy, energy procurement, and regulatory frameworks. Only by combining rigorous measurement and reporting with these multi-layered interventions can the community both quantify the true costs of large-scale AI and identify the most effective levers for reducing its footprint.

However, it is imperative to differentiate between the increasing energy demands of AI and the established workload of traditional datacenter operations. According to the Electric Power Research Institute (EPRI), AI applications currently represent only 10%-20% of electricity consumption of datacenters, meaning the vast majority of consumption is still driven by traditional activities such as cloud computing, streaming, and data retrieval [121]. Data from the International Energy Agency (IEA) supports this conclusion, estimating that while the global electricity consumption of datacenters reached 460 TWh in 2022, this figure could double to over 1000 TWh by 2026 largely due to the rapid scaling of AI workloads [122]. This shift is highlighted by the difference in computational intensity as a single Chat-GPT query is estimated to require 2.9 Wh of electricity, where a standard Google Search consumes only 0.3 Wh. Thus, while traditional infrastructure constitutes the current bulk of the energy footprint, AI is the decisive factor in its projected exponential growth.

IV. Toolboxes, Frameworks, and Emerging Applications

A. Rise of Open-source AI Frameworks

The adoption of open-source AI frameworks has had a transformative impact on the field, promoting the standardization of AI workflows, enhancing scientific reproducibility, and democratizing access to state-of-the-art model development.

1) Definition of AI Frameworks

An AI framework (also known as an ML or DL framework) is a software system that provides the abstractions, libraries, and APIs needed to build, train, evaluate, and deploy AI models. By handling low-level tasks such as tensor operations, automatic differentiation, computational graph optimization, and distribution across hardware such as GPUs or TPUs, AI frameworks allow researchers and engineers to focus on high-level model architecture rather than rebuilding foundational components. As the field has matured, analyses have shown that AI frameworks must balance competing demands, by trading off research flexibility, performance, hardware support, and production readiness [123].

2) Evolution of Major DL Frameworks

The evolution of major DL frameworks follows a shift from early constrained systems such as Theano [124] toward more open, community-driven, and dynamic paradigms. This trend gave rise to PyTorch [125], which supported a dynamic style that offered greater usability and flexibility for research. In contrast, TensorFlow [123] continued to specialize in large-scale industrial applications, leveraging its static computation graph for highly optimized and distributed deployment. Today, ecosystem has become more specialized, with key platforms consolidating their roles. JAX [126] emerged to serve the scientific AI community, offering high-performance differentiable programming for exceptional speed and automatic vectorization. Unifying this diverse landscape is the Hugging Face ecosystem, which functions as a central platform providing model hubs [127] and standardized datasets [128]. This facilitates community-driven fine-tuning and transfer learning, enabling seamless model sharing across all major frameworks. This standardization effect was particularly transformative for NLP, where platforms such as Hugging Face created a unified ecosystem around the transformer architecture [127]. In contrast, computer vision tasks have historically required more specialized data preprocessing and pipelines, making their workflows more difficult for standardization and reproduction.

3) End-to-end Toolchain

Beyond core frameworks, a complete AI pipeline requires a coordinated toolchain to manage the entire model lifecycle, from development to production. This process can be broken down into three key stages as follows.

1) Building and training: in this initial phase, researchers use frameworks such as PyTorch, TensorFlow, and JAX to construct neural architectures and manage training loops. Experiment management and hyperparameter optimization tools are also essential, as reproducible optimization has been shown to reduce outcome variance and improve comparability across experiments.

2) Validation and evaluation: once a model is trained, it must be rigorously validated to ensure generalization, robustness, fairness, and reproducibility. This process extends beyond simple accuracy metrics to include cross-validation, out-of-domain testing, uncertainty quantification, and analysis under distribution shift.

3) Deployment and serving: the final stage converts validated models into production services, which is a task that presents unique challenges for LLMs due to their immense size and computational cost. To make the production services feasible, a suite of optimization techniques are employed. Quantization, for instance, reduces the numerical precision of model weights, drastically cutting the memory footprint and increasing the speed. Other prevalent methods include knowledge distillation [129], where smaller and more efficient student models are trained to replicate the behavior of larger teacher models, and pruning [130], [131], which removes redundant or low-importance parameters to reduce memory footprint and computational cost while preserving the performance of the models.

To manage the high demands of real-time inference, specialized serving frameworks have become essential. Modern engines such as vLLM [132] and SGLang [133] are designed specifically for high-throughput LLM serving. These specialized toolboxes represent a critical evolution from general-purpose model servers, providing the performance necessary for modern generative AI applications. Together, the toolboxes across these pipeline stages form an integrated ecosystem designed to support reproducibility, low-latency inference, versioning, and smooth updates under the realistic constraints of production hardware and environments.

B. Modern Software Stacks

The dramatic shift from simple training loops to modern software stacks has transformed large-scale AI from an experimental slog into an engineering discipline where efficiency is engineered at every layer. Naïve epoch-by-epoch code quickly runs into memory, bandwidth, and latency ceilings. GPUs are idle waiting on small kernels, interconnects become bottlenecks, and energy costs explode, which become conditions that make models with billions of parameters impractical. Modern software stacks expose and exploit device characteristics across layers so that raw compute is converted into usable model capacity at scale.

At the foundation are low-level libraries that provide high-performance primitives and access to hardware features. Libraries such as CUDA and cuDNN expose optimized convolution, reduction, and collective kernels, and the programming model for modern GPUs, removing the need for framework authors to reimplement low-level kernels for each generation of device. Building on these primitives, DL compilers and graph-lowering systems (e. g., XLA [134], TorchScript [135], TVM [77], TensorRT [136]) translate high-level model descriptions into fused, device-tuned kernels and optimized execution graphs, reducing the launch of kernel overhead, improving locality, and enabling cross-operator optimizations that deliver large throughput gains.

Complementing the compilation are parallelization strategies that match the algorithmic structure to hardware. Data parallelism shards minibatches across workers, and when combined with careful scaling and warmup of learning rate, it enables efficient training with very large batch sizes. For models with billions of parameters, parallelism shards weights and activations across devices so that each individual GPU hosts only a fraction of a very large model (e. g., Megatron [137]). Recent systems combine these dimensions and demonstrate training with sizes that are previously unreachievable.

End-to-end optimization closes the deployment loop. Mixed-precision training (FP16/FP32) halves memory bandwidth and compute cost without harming accuracy when it is applied correctly. Quantization, pruning, and activation or weight compression further shrink footprints for inference. And the production runtimes (open neural network exchange (ONNX) runtime [138], DeepSpeed [139], vLLM [132], etc.) operationalize these techniques for high-throughput and low-latency serving. The consequence is that efficiency is now a first-class research problem. Modern stacks and systems-level innovations convert the compute and energy into model capabilities in a far more effective manner, shifting the central question from "can we train this model" to "how cheaply, quickly, and sustainably can we train and serve it". This co-design of software and hardware yields remarkably higher performance per watt and per GPU-hour, enabling experiments and deployments that were previously infeasible.

C. Role of Data in AI's Success

The scale, diversity, and curation of data are often decisive behind modern AI performance. While model architectures and training techniques attract increasing attention, the underlying datasets enable, constrain, and sometimes bias what models can learn. This subsection outlines the historical progression from small datasets to web-scale corpora, highlights open data ecosystems that support reproducibility, examines the data quality challenges, addresses ethical and legal constraints, and presents data-focused trends that shift emphasis from model-first to data-first practices.

1) From Small Datasets to Web-scale Corpora

Early breakthroughs in computer vision and NLP were driven by carefully labeled task-specific datasets such as ImageNet [140] and COCO [141], which enabled systematic benchmarking and architecture-driven improvement. Over the last decade, however, the field shifted toward massive, loosely curated corpora, e.g., Common Crawl [142] for web text, LAION [143] for large-scale image-text pairs, and The Pile [144] for diverse text sources, which now power contemporary large language and multimodal models. The move toward web-scale data has brought dramatic gains through sheer exposure to linguistic and visual variety, but also introduced new issues around noise, redundancy, and provenance.

2) Open Data Ecosystems Supporting Reproducibility

Community-maintained platforms and efforts such as Hugging Face datasets [128] and the BigScience initiative [145] have become central to reproducible research by distributing cleaned, documented datasets and standard loading APIs. These resources lower the barrier of entrance, enable apples-to-apples comparisons, and encourage better dataset versioning and provenance tracking through community curation.

3) Data Quality Challenges: The Hidden Engineering

High-performing models require more than volume. Quality engineering is critical. Common problems include label noise, uninformative or duplicated examples, long-tail distributional gaps, and dataset contamination (e.g., evaluation examples leaking into training), which are issues highlighted as part of the broader engineering risks in deployed ML models. Mitigating these problems requires a toolbox of techniques such as deduplication, filtering, manual vetting, stratified sampling, careful split construction, and substantial compute and human effort. Moreover, biases encoded in source data (demographic, cultural, topical) directly translate into model behavior, making dataset construction and auditing essential engineering tasks rather than afterthoughts [146].

4) Ethical and Legal Constraints

Large-scale data collection raises complex ethical and legal questions. Copyright and ownership of scraped content, consent for personal data, privacy-sensitive information, and the downstream harms of biased or toxic material are all central concerns. Responsible dataset governance demands clear documentation (e. g., datasheets or data cards), provenance tracking, and opt-out and takedown mechanisms with feasible and legal review [146]. Transparent documentation and modular dataset design are practical steps that facilitate safer research and deployment.

5) Data-focused Trends

Recent trends emphasize improving data rather than solely scaling models. Synthetic data generation, active data selection (prioritized sampling and curriculum learning), and targeted human annotation for rare or high-value cases are growing strategies. At the same time, modern generative pipelines increasingly combine these data-centric practices with behavioral and reasoning techniques, e. g., instruction tuning and RLHF to align model behavior, "thinking" or reasoning-augmented methods (chain-of-thought prompting, self-consistency, and tree-of-thoughts) that expose internal deliberation and improve the solving process of complex problem, and inference-time tactics such as retrieval-augmented generation and tool use. Together, synthetic or curated data, active selection, targeted annotation, RLHF, and thinking models form a complementary toolbox that often yields faster, more reliable, and more compute-efficient improvements than blind model scaling, and they slot naturally into the data-focused AI workflow of iterative measurement, targeted correction, and retraining.

D. Emerging Applications in Power Systems

To date, the application of AI in power systems has primarily functioned as an analytical layer focused on observation and prediction. Established techniques such as LSTM networks and random forests are now standard for load and generation predictions and non-intrusive load monitoring. Similarly, CNNs have found widespread application in predictive maintenance, utilizing computer vision to detect faults in transmission lines or thermal anomalies in substation equipment. While these applications have significantly improved observability, the next phase of power system evolution demands a shift from passive analysis to active autonomous control. This transition is being driven by agentic AI, which moves beyond simple data synthesis to autonomous decision-making. In advanced distribution management systems (ADMSs) and virtual power plants (VPPs), agentic AI architectures can replace static optimization tools. Unlike traditional models, these agents operate as orchestrators capable of multi-step reasoning, which interprets natural language goals to coordinate deterministic power flow solvers and economic scheduling algorithms [147], [148]. This could allow for self-healing capabilities where the system can autonomously isolate faults and quickly reconfigure its topology, supporting human operators during critical events. This kind of autonomy requires a fundamental change in how the power system is represented and computed. GNNs are emerging as the superior architecture for digital twins, as they naturally encode the non-Euclidean network topology better than traditional models [149]. Furthermore, the deployment of these advanced models is becoming increasingly decentralized. To mitigate cloud latency in mission-critical assets such as protection relays, algorithms are migrating to the Edge by utilizing the NPUs and low-power FPGAs discussed in Section III to execute inference directly at the smart meter level [150], [151]. This convergence of autonomous agents, topological reasoning, and edge hardware represents the cutting-edge of AI-enabled power systems.

V. Conclusion

A key conclusion of this paper is the explosion of AI technology. With a historical review of AI algorithms, specialized hardware, and improved toolboxes, we have witnessed how the AI community has transformed from its early stages in the 1980s to the state-of-the-art progresses in many scientific domains. DL has enabled technologies with unprecedented capabilities. We must note the acceleration in this domain, e.g., the rise of foundational architectures such as the transformer [15] has radically altered what is feasible. The AI research community has expanded greatly, and open-source platforms along with a far larger group of researchers and developers have turned advances into shared, rapidly reusable artifacts. This is a key reason why AI breakthroughs now compound far faster than in many mature engineering fields.

Despite the recent rapid expansion of capabilities, many contemporary advances in AI are best characterized as engineering refinements and scale-driven improvements rather than fundamentally new theoretical breakthroughs. The predominant trajectory, i. e., bigger models, more parameters, and ever-larger datasets, has yielded impressive empirical gains, but the scale alone is unlikely to constitute a path to AGI. Moreover, much of the effective training signal in current pipelines is synthetic, produced by models themselves. Consequently, new models frequently learn from the outputs of prior models rather than from fresh, independent traces of human cognition. The structured, deliberative patterns of human thought that would most directly advance human-like reasoning remain difficult to obtain at scale, and methods such as RLHF continue to depend on substantial human intervention that does not scale trivially to the continuous supervision a truly general system would require.

Research on hardware and systems will therefore remain indispensable. Continued progress in accelerators, memory hierarchies, and co-design of compiler and toolchain will unlock new levels of energy and cost efficiencies for both AI training and inference, and will enable new deployment modes including on-device and private processing. Nevertheless, efficiency gains are frequently outpaced by rising demand. As AI spreads across products and industries, compute and energy requirements continue to grow. This tension argues for intensified co-design of software and hardware, where software is explicitly developed to leverage hardware features such as sparsity, quantization, and model partitioning, and hardware is engineered to support the algorithmic patterns that materially improve the efficiency.

Looking beyond current technologies and architectures, compute-in-memory (CIM) and quantum computing stand out as incoming technologies with the potential to fundamentally shift the AI hardware paradigm. CIM addresses the critical energy costs of data movement by performing calculations directly within memory arrays. This offers a way to bypass the physical traffic jam between the processor and storage that usually slows down computations [152]. Simultaneously, quantum machine learning (QML) promises to revolutionize the algorithmic efficiency by handling high-dimensional optimization landscapes and probabilistic sampling tasks that are currently impossible for classical hardware [153]. While these technologies are not yet certain replacements for standard hardware, they represent leaps that could redefine the efficiency and capability of future AI systems.

Ecosystems and toolchains are likewise consolidating toward integrated, modular stacks. A prevailing trend is the emergence of comprehensive platforms that unite model repositories, dataset registries, evaluation suites, fine-tuning pipelines, monitoring, and safety tooling, mostly with open-source implementations. Using small, specialized AI models with tools (such as retrieval and agents) facilitate to build, check, and change the system. It also boosts reuse, letting teams quickly update shared models and data.

Several research directions merit particular emphasis. Multimodality, lifelong learning, and causal and compositional reasoning are likely to be more consequential than further increases in parameter count. Scalable alternatives to intensive human supervision, i.e., improved preference learning, active learning, scalable self-play, higher-fidelity simulation, and more principled synthetic-data generation, will be critical to reduce dependence on costly manual labeling. Equally essential are robust evaluation frameworks, interpretability and runtime monitoring tools, and mechanisms that permit safe, accountable deployment across domains.

A final and equally important topic is to regulate the application of AI systems. The total energy consumed by AI not only depends on hardware efficiency, but also on how often and how heavily users invoke the AI services. Therefore, policies and design choices that promote energy-aware usage are essential for any effort to reduce emissions. In practice, this means setting default options that favor smaller, faster models for everyday tasks, helping users understand when large, high-capacity models are actually needed, and making it easy to switch to lower-energy alternatives when performance differences are minimal. These demand-side measures mirror familiar conservation habits in other areas, where small individual choices collectively have a massive impact. Embedding energy awareness into product design, deployment policies, and public guidance complements the technical work being done at the hardware, software, and power system levels.

In summary, future progress will be a co-evolution of software, hardware, toolboxes, and community practices. Scaling will remain an important lever, but achieving more general, robust, and human-aligned intelligence will require better data (not merely more data), smarter algorithms, hardware-aware designs, richer tooling and evaluation, and sustained interdisciplinary engagement with multiple scientific domains. Only through such an integrated program can technological advances be aligned with the supply of low-carbon electricity, as well as the broader goals of society for safety, fairness, and resilience.

References

[1] P. S. Huang, X. He, J. Gao *et al*., “Learning deep structured semantic models for web search using clickthrough data,” in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, San Francisco, USA, Oct. 2013, pp. 2333-2338.

[2] X. He, L. Liao, H. Zhang *et al.* (2017, Aug.). Neural collaborative filtering. [Online]. Available: https://arxiv.org/abs/1708.05031

[3] O. Vinyals and Q. Le. (2015, Jul.). A neural conversational model. [Online]. Available: https://arxiv.org/abs/1506.05869

[4] M. Bojarski, D. D. Testa, D. Dworakowski *et al.* (2016, Apr.). End to end learning for self-driving cars. [Online]. Available: https://arxiv.org/abs/1604.07316

[5] T. B. Brown, B. Mann, N. Ryder *et al.* (2020, Jul.). Language models are few-shot learners. [Online]. Available: https://arxiv. org/abs/2005.14165

[6] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, Jun. 2019, pp. 4396-4405.

[7] D. Silver, A. Huang, C. J. Maddison *et al*., “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, pp. 484-489, Jan. 2016.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, May 2017.

[9] J. Devlin, M.-W. Chang, K. Lee *et al.* (2019, May). BERT: pre-training of deep bidirectional transformers for language understanding. [Online]. Available: https://arxiv.org/abs/1810.04805

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[13] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proceedings of 2013 IEEE International Conference on Acoustics*, *Speech and Signal Processing*, Vancouver, Canada, May 2013, pp. 6645-6649.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[15] A. Vaswani, N. Shazeer, N. Parmar *et al.* (2023, Feb.). Attention is all you need. [Online]. Available: https://arxiv.org/abs/1706.03762

[16] S. Grigorescu, B. Trasnea, T. Cocias *et al*., “A survey of deep learning techniques for autonomous driving,” *Journal of Field Robotics*, vol. 37, no. 3, pp. 362-386, Apr. 2020.

[17] A. Radford and K. Narasimhan. (2018, Dec.). Improving language understanding by generative pre-training. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245

[18] A. B. Nassif, M. A. Talib, Q. Nasir *et al*., “Machine learning for anomaly detection: a systematic review,” *IEEE Access*, vol. 9, pp. 78658-78700, May 2021.

[19] C. Sun, A. Shrivastava, S. Singh *et al*., “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, Italy, Oct. 2017, pp. 843-852.

[20] C. Bogmans, P. Gomez-Gonzalez, G. Ganpurev *et al.* (2025, Mar.). Power hungry. [Online]. Available: https://www.imf.org/en/publications/wp/issues/2025/04/21/power-hungry-how-ai-will-drive-energy-demand-566304

[21] Z. Fan, Z. Yan, and S. Wen, “Deep learning and artificial intelligence in sustainability: a review of SDGs, renewable energy, and environmental health,” *Sustainability*, vol. 15, no. 18, p. 13493, Sept. 2023.

[22] W. van Melle, “MYCIN: a knowledge-based consultation program for infectious disease diagnosis,” *International Journal of Man-Machine Studies*, vol. 10, no. 3, pp. 313-322, May 1978.

[23] M. I. Jordan and T. M. Mitchell, “Machine learning: trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255-260, Jul. 2015.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533-536, Oct. 1986.

[25] A. F. Agarap. (2018, Mar.). Deep learning using rectified linear units (ReLU). [Online]. Available: https://arxiv.org/abs/1803.08375

[26] K. Fukushima and S. Miyake, “Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognition*, vol. 15, no. 6, pp. 455-469, Jan. 1982.

[27] Y. LeCun, L. Bottou, Y. Bengio *et al*., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, May 2017.

[29] K. Cho, B. van Merrienboer, C. Gulcehre *et al.* (2014, Jun.). Learning phrase representations using RNN encoder-decoder for statistical machine translation. [Online]. Available: https://arxiv.org/abs/1406.1078

[30] K. He, X. Zhang, S. Ren *et al*., “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, Jun. 2016, pp. 770-778.

[31] C. Szegedy, W. Liu, Y. Jia *et al*., “Going deeper with convolutions,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, Jun. 2015, pp. 1-9.

[32] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, Jul. 2015, pp. 448-456.

[33] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, Oct. 2015, pp. 234-241.

[34] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, Jun. 2014, pp. 1764-1772.

[35] X. He, K. Deng, X. Wang *et al.* (2020, Feb.). LightGCN: simplifying and powering graph convolution network for recommendation. [Online]. Available: https://arxiv.org/abs/2002.02126

[36] J. Gilmer, S. S. Schoenholz, P. F. Riley *et al*., “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, Aug. 2017, pp. 1263-1272.

[37] A. Radford, J. W. Kim, C. Hallacy *et al.* (2021, Jul.). Learning transferable visual models from natural language supervision. [Online]. Available: https://arxiv.org/abs/2103.00020

[38] J. Ho, A. Jain, and P. Abbeel. (2020, Jun.). Denoising diffusion probabilistic models. [Online]. Available: https://arxiv.org/abs/2006.11239

[39] R. Rombach, A. Blattmann, D. Lorenz *et al*., “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, Jun. 2022, pp. 10674-10685.

[40] G. Team, R. Anil, S. Borgeaud *et al.* (2023, Dec.). Gemini: a family of highly capable multimodal models. [Online]. Available: https://arxiv.org/abs/2312.11805

[41] OpenAI. (2023, Dec.). GPT-4V(ision) system card. [Online]. Available: https://openai.com/index/gpt-4v-system-card/

[42] S. Zhang, L. Dong, X. Li *et al.* (2023, Aug.). Instruction tuning for large language models: a survey. [Online]. Available: https://arxiv.org/abs/2308.10792

[43] H. Liu, C. Li, Q. Wu *et al.* (2023, Apr.). Visual instruction tuning. [Online]. Available: https://arxiv.org/abs/2304.08485

[44] L. Ouyang, J. Wu, X. Jiang *et al.* (2022, Mar.). Training language models to follow instructions with human feedback. [Online]. Available: https://arxiv.org/abs/2203.02155

[45] J. Kaplan, S. McCandlish, T. Henighan *et al.* (2020, Jan.). Scaling laws for neural language models. [Online]. Available: https://arxiv.org/abs/2001.08361

[46] J. Hoffmann, S. Borgeaud, A. Mensch *et al.* (2022, Mar.). Training compute-optimal large language models. [Online]. Available: https://arxiv.org/abs/2203.15556

[47] DeepSeek-AI. (2025, Nov.). DeepSeek-V3 technical report. [Online]. Available: https://github.com/deepseek-ai/DeepSeek-V3

[48] Meta AI. (2025, Nov.). The Llama 4 herd: the beginning of a new era of natively multimodal AI innovation. [Online]. Available: https://ai.meta.com/blog/meta-llama-4-herd/

[49] K. Sun, D. Zheng, and Q. Lu, "A simulation study of OBDD-based proper splitting strategies for power systems under consideration of transient stability," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 389-399, Feb. 2005.

[50] C. Wang, B. Zhang, Z. Hao *et al.*, "A real-time searching method for splitting surfaces of the power system," *Proceedings of the CSEE*, vol. 30, no. 7, pp. 48-55, Mar. 2010.

[51] Q. Team. (2025, Sept.). Qwen3-Max: just scale it. [Online]. Available: https://arxiv.org/abs/2408.18974

[52] A. Radford, J. Wu, R. Child *et al.* (2019, Feb.). Language models are unsupervised multitask learners. [Online]. Available: https://openai.com/research/better-language-models

[53] T. B. Brown, B. Mann, N. Ryder *et al.* (2020, May). Language models are few-shot learners. [Online]. Available: https://arxiv. org/abs/2005.14165

[54] B. Shen, Q. Li, B. Chen *et al.*, "Large language model-based security situation awareness for smart grid: framework and approaches," *IEEE Access*, vol. 13, pp. 173600-173613, Oct. 2025.

[55] C. T. Hazera, M. I. Ibrahem, and M. M. Fouda, "Few-shot learning for CPS anomaly detection: a survey on smart grid applications," in *Proceedings of the 2025 1st International Conference on Secure IoT, Assured and Trusted Computing*, Dayton, USA, Feb. 2025, pp. 1-7.

[56] D. Podell, Z. English, K. Lacey *et al.* (2025, Jan.). SDXL: improving latent diffusion models for high-resolution image synthesis. [Online]. Available: https://arxiv.org/abs/2307.01952

[57] A. Ramesh, M. Pavlov, G. Goh *et al.* (2021, Feb.). Zero-shot text-to-image generation. [Online]. Available: https://arxiv.org/abs/2102.12092

[58] A. Kumar, "Building autonomous AI agents based AI infrastructure," *International Journal of Computer Trends and Technology*, vol. 72, no. 11, pp. 116-125, Nov. 2024.

[59] J. Wang, J. Wang, B. Athiwaratkun *et al.* (2024, Jul.). Mixture-of-agents enhances large language model capabilities. [Online]. Available: https://arxiv.org/abs/2406.04692

[60] S. Yuan, K. Song, J. Chen *et al.* (2024, Jun.). EvoAgent: towards automatic multi-agent generation via evolutionary algorithms. [Online]. Available: https://arxiv.org/abs/2406.10580

[61] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: autonomous intelligence for complex goals – a comprehensive survey," *IEEE Access*, vol. 13, pp. 18912-18936, Jan. 2025.

[62] C. Ma, J. Li, K. Wei *et al.*, "Trusted AI in multiagent systems: an overview of privacy and security for distributed learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1097-1132, Sept. 2023.

[63] X. Lin, Y. Ning, J. Zhang *et al.* (2025, Jan.). LLM-based agents suffer from hallucinations: a survey of taxonomy, methods, and directions. [Online]. Available: https://arxiv.org/abs/2501.15029

[64] S. Mittal and J. S. Vetter, "A survey of CPU-GPU heterogeneous computing techniques," *ACM Computing Surveys*, vol. 47, no. 4, pp. 1-35, Jul. 2015.

[65] NVIDIA Corporation. (2017, Aug.). NVIDIA Tesla V100 GPU architecture. [Online]. Available: https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

[66] H. Wong, M. M. Papadopoulou, M. Sadooghi-Alvandi *et al.*, "Demystifying GPU microarchitecture through microbenchmarking," in *Proceedings of the 2010 IEEE International Symposium on Performance Analysis of Systems & Software*, White Plains, USA, Mar. 2010, pp. 235-246.

[67] S. Mittal and J. S. Vetter, "A survey of methods for analyzing and improving GPU energy efficiency," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1-23, Jan. 2015.

[68] J. Nickolls, I. Buck, M. Garland *et al.*, "Scalable parallel programming with CUDA," *Queue*, vol. 6, no. 2, pp. 40-53, Mar. 2008.

[69] A. Munshi, "The OpenCL specification," in *Proceedings of the 2009 IEEE Hot Chips 21 Symposium*, Stanford, USA, Aug. 2009, pp. 1-314.

[70] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Boston: Addison-Wesley Professional, 2010.

[71] J. Nickolls, I. Buck, M. Garland *et al.*, "Scalable parallel programming with CUDA," in *Proceedings of ACM SIGGRAPH 2008 Classes*, Los Angeles, USA, Aug. 2008, pp. 1-14.

[72] K. Fatahalian, J. Sugerman, and P. Hanrahan, "Understanding the efficiency of GPU algorithms for matrix-matrix multiplication," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, Grenoble, France, Aug. 2004, pp. 133-137.

[73] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Canada, Jun. 2009, pp. 873-880.

[74] JEDEC Solid State Technology Association. (2021, Aug.). High bandwidth memory (HBM3) DRAM. [Online]. Available: https://www.jedec.org/standards-documents/docs/jesd238a

[75] S. Markidis, S. W. Der Chien, E. Laure *et al.*, "NVIDIA tensor core programmability, performance & precision," in *Proceedings of the 2018 IEEE International Parallel and Distributed Processing Symposium Workshops*, Vancouver, Canada, May 2018, pp. 522-531.

[76] D. Narayanan, M. Shoeybi, J. Casper *et al.*, "Efficient large-scale language model training on GPU clusters using megatron-LM," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, St. Louis, USA, Nov. 2021, pp. 1-15.

[77] T. Chen, T. Moreau, Z. Jiang *et al.* (2018, May). TVM: an automated end-to-end optimizing compiler for deep learning. [Online]. Available: https://arxiv.org/abs/1802.04799

[78] OpenXLA Project. (2024, Mar.). XLA (accelerated linear algebra). [Online]. Available: https://github.com/openxla/xla

[79] J. You, J.-W. Chung, and M. Chowdhury, "Zeus: understanding and optimizing GPU energy consumption of DNN training," in *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation*, Boston, USA, Apr. 2023, pp. 119-139.

[80] N. Jegham, M. Abdelatti, L. Elmoubarki *et al.* (2025, Feb.). How hungry is AI? Benchmarking energy, water, and carbon footprint of LLM inference. [Online]. Available: https://arxiv.org/abs/2502.06799

[81] N. P. Jouppi, C. Young, N. Patil *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture*, Toronto, Canada, Jun. 2017, pp. 1-12.

[82] N. Jouppi, G. Kurian, S. Li *et al.*, "TPU v4: an optically reconfigurable supercomputer for machine learning with hardware support for embeddings," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando, USA, Jun. 2023, pp. 1-14.

[83] Amazon Web Services. (2025, Jan.). Trainium architecture – AWS neuron documentation. [Online]. Available: https://awsdocs-neuron.readthedocs-hosted.com/en/latest/general/arch/aws-trainium-arch.html

[84] Amazon Web Services. (2025, Jan.). Inferentia architecture – AWS neuron documentation. [Online]. Available: https://awsdocs-neuron.readthedocs-hosted.com/en/latest/general/arch/aws-inferentia-arch.html

[85] R. Borkar, A. Wall, P. Pulavarthi *et al.* (2024, May). Azure Maia for the era of AI: from silicon to software to systems. [Online]. Available: https://arxiv.org/abs/2410.05277

[86] P.-S. V. Sun, A. Titterton, A. Gopiani *et al.* (2022, Nov.). Intelligence processing units accelerate neuromorphic learning. [Online]. Available: https://arxiv.org/abs/2211.06248

[87] Cerebras Systems. (2025, Mar.). The future of AI is wafer scale: wafer scale engine 3 (WSE-3). [Online]. Available: https://www.cerebras.net/wafer-scale-engine/

[88] Intel Corporation. (2024, Mar.). Intel Gaudi 3 AI accelerator white paper. [Online]. Available: https://www.intel.com/content/www/us/en/content-details/817486/intel-gaudi-3-ai-accelerator-architecture-white-paper.html

[89] K. J. Lee, "Architecture of neural processing unit for deep neural networks," *Advances in Computers*, vol. 122, pp. 217-245, Mar. 2021.

[90] A. Orhon, A. Wadhwa, Y. Kim *et al.* (2022, Jul.). Deploying transformers on the apple neural engine. [Online]. Available: https://arxiv. org/abs/2207.01787

[91] K. L. Loh. (2024, Jun.). Unlocking on-device generative AI with an NPU and heterogeneous computing. [Online]. Available: https://developer.apple.com/videos/play/wwdc2024/10047/

[92] K. Seshadri, B. Akin, J. Laudon *et al.* (2022, Nov.). An evaluation of edge TPU accelerators for convolutional neural networks. [Online]. Available: https://arxiv.org/abs/2211.06406

[93] C. Song, C. Ye, Y. Sim *et al.*, "Hardware for deep learning acceleration," *Advanced Intelligent Systems*, vol. 6, no. 10, p. 2300762, Oct. 2024.

[94] C. Silvano, D. Ielmini, F. Ferrandi *et al.*, "A survey on deep learning hardware accelerators for heterogeneous HPC platforms," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1-39, Nov. 2025.

[95] E. Talpes, D. Das Sarma, G. Venkataramanan *et al.*, "Compute solution for Tesla's full self-driving computer," *IEEE Micro*, vol. 40, no. 2, pp. 25-35, Mar. 2020.

[96] A. Firoozshahian, J. Coburn, R. Levenstein *et al.*, "MTIA: first generation silicon targeting meta's recommendation systems," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando, USA, Jun. 2023, pp. 1-13.

[97] F. Yan, A. Koch, and O. Sinnen. (2024, Jan.). A survey on FPGA-based accelerator for ML models. [Online]. Available: https://arxiv.org/abs/2401.03436

[98] Advanced Micro Devices, Inc. (AMD). (2020, Oct.). Versal: the first adaptive compute acceleration platform (ACAP). [Online]. Available: https://www. amd. com/content/dam/amd/en/documents/adaptable-solution-docs/white-papers/versal-acap-wp505.pdf

[99] J. Fowers, K. Ovtcharov, M. Papamichael *et al.*, "A configurable cloud-scale DNN processor for real-time AI," in *Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture*, Los Angeles, USA, Jun. 2018, pp. 1-14.

[100] B. Vogginger, A. Rostami, V. Jain *et al.* (2024, May). Neuromorphic hardware for sustainable AI data centers. [Online]. Available: https://arxiv.org/abs/2405.04719

[101] A. Shrestha, H. Fang, Z. Mei *et al.*, "A survey on neuromorphic computing: models and hardware," *IEEE Circuits and Systems Magazine*, vol. 22, no. 2, pp. 6-35, Apr. 2022.

[102] T. Luo, W. Wong, R. S. M. Goh *et al.*, "Achieving green AI with energy-efficient deep learning using neuromorphic computing," *Communications of the ACM*, vol. 66, no. 7, pp. 52-57, Jul. 2023.

[103] B. V. Benjamin, P. Gao, E. McQuinn *et al.*, "Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699-716, May 2014.

[104] Human Brain Project (HBP). (2023, Apr.). Neuromorphic computing. [Online]. Available: https://www. humanbrainproject. eu/en/science/explore-the-brain/brain-inspiration/neuromorphic-computing/

[105] C.-J. Wu, R. Raghavendra, U. Gupta *et al.* (2022, Dec.). Sustainable AI: environmental implications, challenges and opportunities. [Online]. Available: https://arxiv.org/abs/2111.00364

[106] S. Bisson. (2024, Mar.). Inside today's Azure AI cloud data centers. [Online]. Available: https://www. infoworld. com/article/3715266/inside-todays-azure-ai-cloud-data-centers.html

[107] NVIDIA Corporation. (2024, Jun.). NVIDIA quantum, spectrum, and LinkX product lines. [Online]. Available: https://www. nvidia. com/en-us/networking/products/data-center-connectivity/

[108] NVIDIA Corporation. (2024, Jan.). NVLink & NVSwitch for advanced multi-GPU communication. [Online]. Available: https://www.nvidia.com/en-us/data-center/nvlink/

[109] CXL Consortium. (2024, Aug.). Compute express link (CXL) specification version 3.0. [Online]. Available: https://www. computeexpresslink.org/download-the-specification

[110] OpenSFS. (2024, Jan.). Lustre documentation. [Online]. Available: https://doc.lustre.org/

[111] IBM Corporation. (2024, Apr.). GPFS architecture. [Online]. Available: https://www. ibm. com/docs/en/spectrum-scale/5.1.9? topic=overview-gpfs-architecture

[112] NVIDIA Corporation. (2024, May). DGX SuperPOD: AI infrastructure for enterprise deployments. [Online]. Available: https://www. nvidia. com/en-us/data-center/dgx-superpod/

[113] Google Cloud. (2024, Mar.). TPU v4. [Online]. Available: https://cloud.google.com/tpu/docs/system-architecture-tpu-vm

[114] Amazon Web Services. (2024, May). Amazon EC2 P4 instances. [Online]. Available: https://aws.amazon.com/ec2/instance-types/p4/

[115] Amazon Web Services. (2024, Nov.). Amazon EC2 P5 instances. [Online]. Available: https://aws.amazon.com/ec2/instance-types/p5/

[116] Microsoft Corporation. (2024, Mar.). ND family virtual machine size series. [Online]. Available: https://learn.microsoft.com/en-us/azure/virtual-machines/nd-series

[117] C. Gordon. (2024, Mar.). ChatGPT and generative AI innovations are creating sustainability. [Online]. Available: https://www.forbes.com/sites/calumgordon/2024/03/19/chatgpt-and-generative-ai-innovations-are-creating-sustainability-havoc/

[118] S. del Rey, L. Cruz, X. Franch *et al.* (2025, Jan.). Estimating deep learning energy consumption based on model architecture and training environment. [Online]. Available: https://arxiv.org/abs/2501.08339

[119] F. Oviedo, F. Kazhamiaka, E. Choukse *et al.* (2025, Sept.). Energy use of AI inference: efficiency pathways and test-time compute. [Online]. Available: https://arxiv.org/abs/2509.03756

[120] S. Chen, "How much energy will AI really consume? The good, the bad and the unknown," *Nature*, vol. 639, no. 8053, pp. 22-24, Mar. 2025.

[121] Electric Power Research Institute. (2024, Jun.). Powering intelligence: analyzing artificial intelligence and data center energy consumption. [Online]. Available: https://www.epri. com/research/products/000000003002028905

[122] International Energy Agency. (2024, Jan.). Electricity 2024: analysis and forecast to 2026. [Online]. Available: https://www. iea. org/reports/electricity-2024

[123] M. Abadi, A. Agarwal, P. Barham *et al.* (2016, Mar.). TensorFlow: a system for large-scale machine learning. [Online]. Available: https://arxiv.org/abs/1603.04467

[124] The Theano Development Team *et al.* (2016, Nov.). Theano: a Python framework for fast computation of mathematical expressions. [Online]. Available: https://arxiv.org/abs/1605.02688

[125] A. Paszke, S. Gross, F. Massa *et al.* (2019, Dec.). PyTorch: an imperative style, high-performance deep learning library. [Online]. Available: https://arxiv.org/abs/1912.01703

[126] J. Bradbury, R. Frostig, P. Hawkins *et al.* (2018, Dec.). JAX: composable transformations of Python+NumPy programs. [Online]. Available: https://github.com/google/jax/blob/main/README.md

[127] T. Wolf, L. Debut, V. Sanh *et al.*, "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2020, pp. 38-45.

[128] Q. Lhoest, A. V. del Moral, Y. Jernite *et al.*, "Datasets: a community library for natural language processing," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, Nov. 2021, pp. 175-184.

[129] G. Hinton, O. Vinyals, and J. Dean. (2015, Mar.). Distilling the knowledge in a neural network. [Online]. Available: https://arxiv. org/abs/1503.02531

[130] S. Han, J. Pool, J. Tran *et al.* (2015, Oct.). Learning both weights and connections for efficient neural networks. [Online]. Available: https://arxiv.org/abs/1506.02626

[131] E. Frantar and D. Alistarh. (2023, Jan.). SparseGPT: massive language models can be accurately pruned in one-shot. [Online]. Available: https://arxiv.org/abs/2301.00774

[132] W. Kwon, Z. Li, S. Zhuang *et al.* (2023, Jun.). Efficient memory management for large language model serving with PagedAttention. [Online]. Available: https://arxiv.org/abs/2309.06180

[133] L. Zheng, L. Yin, Z. Xie *et al.* (2024, Jun.). SGLang: efficient execution of structured language model programs. [Online]. Available: https://arxiv.org/abs/2406.07498

[134] Google. (2024, Mar.). XLA: accelerated linear algebra. [Online]. Available: https://www.tensorflow.org/xla

[135] A. Paszke, S. Gross, F. Massa *et al.* (2019, Aug.). PyTorch: an imperative style, high-performance deep learning library. [Online]. Available: https://pytorch.org/

[136] NVIDIA Corporation. (2024, Jul.). NVIDIA TensorRT documentation. [Online]. Available: https://docs.nvidia.com/deeplearning/tensorrt/

[137] M. Shoeybi, M. Patwary, R. Puri *et al.* (2019, Aug.). Megatron-LM: training multi-billion parameter language models using model parallelism. [Online]. Available: https://arxiv.org/abs/1909.08053

[138] ONNX Runtime developers. (2024, May). ONNX runtime. [Online]. Available: https://onnxruntime.ai/

[139] J. Rasley, S. Rajbhandari, O. Ruwase *et al.*, "DeepSpeed: system optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, virtual event, Aug. 2020, pp. 3505-3506.

[140] J. Deng, W. Dong, R. Socher *et al.*, "ImageNet: a large-scale hierarchical image database," in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, Jun. 2009, pp. 248-255.

[141] T.-Y. Lin, M. Maire, S. Belongie *et al.* (2015, May). Microsoft CO-

CO: common objects in context. [Online]. Available: https://arxiv.org/abs/1405.0312

[142] Common Crawl Foundation. (2024, Mar.). Common Crawl: open repository of web crawl data. [Online]. Available: https://commoncrawl.org/

[143] C. Schuhmann, R. Beaumont, R. Vencu *et al.* (2022, Oct.). LAION-5B: an open large-scale dataset for training next generation image-text models. [Online]. Available: https://arxiv.org/abs/2210.08402

[144] L. Gao, S. Biderman, S. Black *et al.* (2020, Dec.). The pile: an 800GB dataset of diverse text for language modeling. [Online]. Available: https://arxiv.org/abs/2101.00027

[145] BigScience Workshop *et al.* (2023, Mar.). BLOOM: a 176B-parameter open-access multilingual language model. [Online]. Available: https://arxiv.org/abs/2211.05100

[146] T. Gebru, J. Morgenstern, B. Vecchione *et al.* (2021, Mar.). Datasheets for datasets. [Online]. Available: https://arxiv.org/abs/1803.09010

[147] S. Ghosh and G. Mittal. (2025, Feb.). Agentic AI systems in electrical power systems engineering: current state-of-the-art and challenges. [Online]. Available: https://arxiv.org/abs/2502.18142

[148] H. Jin, K. Kim, and J. Kwon, "GridMind: LLMs-powered agents for power system analysis and operations," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, St. Louis, USA, Nov. 2025, pp. 560-568.

[149] I. Zacarias, O. B. Taarit, and A. Jukan. (2025, Jan.). On effectiveness of graph neural network architectures for network digital twins (NDTs). [Online]. Available: https://arxiv.org/abs/2501.05801

[150] S. Liang, S. Jin, and Y. Chen, "A review of edge computing technology and its applications in power systems," *Energies*, vol. 17, no. 13, p. 3230, Jul. 2024.

[151] Y. Himeur, A. N. Sayed, A. Alsalemi *et al.*, "Edge AI for Internet of energy: challenges and perspectives," *Internet of Things*, vol. 25, p. 101035, Apr. 2024.

[152] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh *et al.*, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, no. 7, pp. 529-544, Jul. 2020.

[153] J. Biamonte, P. Wittek, N. Pancotti *et al.*, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195-202, Sept. 2017.

Angelos Vlachos received the B.S. and M.S. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2023. From 2024 to 2025, he worked as a Research Assistant with the Computer Systems Laboratory (CSLab) and the Artificial Intelligence and Systems Laboratory (AILS), National Technical University of Athens (NTUA), Athens, Greece. He is currently pursuing the Ph.D. degree in multimodal artificial intelligence at AILS, NTUA. His research interests include multimodal machine learning and artificial intelligence, especially visual-language understanding, vision-and-language reasoning, and machine learning system such as neural network optimization.

Anastasia Poulopoulou is pursuing the Diploma (joint) degree in electrical and computer engineering at the National Technical University of Athens, Athens, Greece. Her research interests include high-performance computing, parallel processing, and computer system.

Christina Giannoula received the Ph. D. degree from School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2022. She is currently a Tenure-track Faculty (Assistant Professor level) at the Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany. Before that, she was a Postdoctoral Researcher at the University of Toronto, Toronto, Canada. Her research interests include intersection of computer architecture, computer system, and high-performance computing.

Georgios Goumas received the Ph.D. degree from the School of Electrical and Computer Engineering (ECE), National Technical University of Athens (NTUA), Athens, Greece, in 2004, where he is currently a Professor. He is also a Senior Researcher at the Computer Systems Laboratory (CSLab) of NTUA. His research interests include high-performance computing and architecture, cloud computing, resource allocation policy, resource-demanding application, sparse algebra, automatic parallelizing compiler, and parallel programming model.

Nectarios Koziris is a Professor of computer science and the former Dean of the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece. From 2019, he is serving as the National Delegate at the European High Performance Joint Undertaking (EuroHPC JU), where he is a Member of the Governing Board. In the past twenty-five years, he initiated and designed several large-scale computing system infrastructures in Greece and the European Union (EU), including cloud, network, and supercomputer, and recently the Daedalus supercomputer. His research interests include high-performance and large-scale computing system and architecture, and parallel and distributed computing.